The last mile delivery problem

Yuan Yuan, Diego Cattaruzza, Maxime Ogier, Frédéric Semet
Univ. Lille, CNRS, Centrale Lille, Inria,
UMR 9189 - CRIStAL Lille, France

Email: frederic.semet@centralelille.fr

1 Introduction and problem definition

E-commerce is a thriving market around the world and suits very well the busy lifestyle of today's customers. An annual survey by analytics firm comScore and UPS revealed that consumers in US were purchasing more things online than in stores in 2016[1]. According to the Ecommerce fondation, 1.4 billion people purchased goods and/or services online at least once in 2015. They spent \$ 2,272.7 billion online, which results in an average spending per e-shopper of \$ 1,582. It is obvious that the e-commerce growth poses a huge challenge for transportation companies, especially in the last mile delivery. According to [1], the last mile parcel delivery cost often reaches or even exceeds 50% of the total transportation cost, making it a top concern for many companies.

Nowadays, the most common last mile delivery service is home delivery. Customers wait at home to get their orders. Besides home delivery, companies like Amazon and Fedex, develop locker and pick-up&go delivery services. When customers shop online, they can choose a nearby locker or a store offering a pick-up&go counter. In the past two years, a new concept called *trunk delivery*, has been proposed. Here, customers' orders can be delivered to the trunk of their cars. Volvo launched its in-car delivery service in Sweden in 2016. The courier has a one-time digital code to get access to the car.

Trunk delivery is different from home delivery and locker delivery since the car moves during the day and can be in different locations during the planning horizon. We study an efficient last mile delivery system that combines all these delivery services: home, locker, pick-up&go location and car trunk.

In this presentation, we address the routing problem for one vehicle. The problem is modeled on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. The set of vertices $\mathcal{V} = \{0, 1, ..., N\}$ is partitioned into $\mathcal{C}_0 = \{0\}, \mathcal{C}_1, ..., \mathcal{C}_K$ clusters. Cluster $\mathcal{C}_0$ contains only the depot. Each other cluster $\mathcal{C}_k$, $k > 0$ represents the set of alternative locations on which client k can

[1] `http://fortune.com/2016/06/08/online-shopping-increases/`

be delivered. Each vertex is associated with a time-window $[E_i, L_i], i \in \{0, 1, ..., N\}$ with $[E_0, L_0] = [0, T]$. A visit can only be made to a vertex during its time-window and an early arrival leads to waiting time while a late arrival causes infeasibility. In a cluster, the time-windows associated with the vertices may overlap. Arcs are only defined between vertices belonging to different clusters, that is, $\mathcal{A} = \{(i, j) : i \in \mathcal{C}_k, j \in \mathcal{C}_l, k \neq l\}$. Each arc $(i, j) \in \mathcal{A}$ is associated with a traveling cost C_{ij} and time T_{ij}. A tour starts from and ends at the depot.

The objective is to find a minimum cost tour visiting each customer at one location within the associated time-window. The problem that arises is called the Generalized Traveling Salesman Problem with Time Windows (GTSPTW). We assume that this problem is static and deterministic, namely all customer locations and the associated time-windows are known with certainty in advance. The GTSPTW can be modeled as follows.

$$\min \sum_{(i,j)\in\mathcal{A}} C_{ij}x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{i\in\mathcal{C}_k} y_i = 1 \qquad k \in \{0, 1, ..., K\}, \tag{2}$$

$$\sum_{(i,j)\in\delta^+(i)} x_{ij} = \sum_{(j,i)\in\delta^-(i)} x_{ji} = y_i \qquad \forall i \in \mathcal{V}, \tag{3}$$

$$\sum_{i\in\mathcal{C}_k} E_i y_i \leq t_k \leq \sum_{i\in\mathcal{C}_k} L_i y_i \qquad k \in \{0, 1, ..., K\}, \tag{4}$$

$$t_k - t_\ell + T_{ij}x_{ij} \leq \sum_{u\in\mathcal{C}_k} L_u y_u - \sum_{v\in\mathcal{C}_\ell} E_v y_v - (L_i - E_j)x_{ij} \quad \forall (i, j) \in \mathcal{A}, i \in \mathcal{C}_k, j \in \mathcal{C}_\ell, \tag{5}$$

$$y_i \in \{0, 1\} \qquad \forall i \in \mathcal{V}, \tag{6}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in \mathcal{A}. \tag{7}$$

where $\delta^+(i) = \{(i, j) \in \mathcal{A}, j \neq i\}$ and $\delta^-(i) = \{(j, i) \in \mathcal{A}, j \neq i\}$. The objective function (1) minimizes the overall costs. Constraints (2) ensure that exactly one vertex from each cluster is visited. Constraints (3) are flow conservation constraints. Constraints (4) ensure that a vertex is visited during its time-window. Constraints (5) ensure that the arrival and traveling times are consistent, meanwhile eliminating subtours. Constraints (6) and (7) are variable definitions.

When time-windows are not considered, the GTSPTW reduces to the well-known

Generalized Travelling Salesman Problem (Fischetti et al. [2]). For the multi-vehicle case, the problem is named Generalized Vehicle Routing Problem with TW (Moccia et al. [4]). The special case where time-windows on clients do not overlap has been recently considered by Reyes et al. [3]. The problem is called the Vehicle Routing Problem (VRP) with Roaming Deliveries and models the case when only trunk deliveries are considered.

2 A branch-and-cut scheme

We propose several valid inequalities for GTSPTW. Some of them are derived from valid inequalities for the asymmetric traveling salesman problem with time-windows, others are specific. The main constraints are the followings :

- Feasible path inequality. $x_{ij} + \sum_{h \in \mathcal{S}_{ij}} y_h \geq y_i + y_j - 1 \quad i \in \mathcal{C}_k, j \in \mathcal{C}_\ell$,
  where $\mathcal{S}_{ij} = \{h \in \mathcal{V} \setminus (\mathcal{C}_k \cup \mathcal{C}_\ell) | E_i + T_{ih} \leq L_h, E_h + T_{hj} \leq L_j, E_i + T_{ih} + T_{hj} \leq L_j\}$ for $i \in \mathcal{C}_k$ and $j \in \mathcal{C}_\ell$.
  Two vertices are visited either directly or there exists a connection between them.

- Infeasible path elimination inequality. Several infeasible path elimination constraints can be defined. The simplest ones are the followings:
  $x_{ij} + \sum_{h \in \mathcal{C}_k^{ij}} y_h \leq 1, \quad k \in \{1, \ldots, K\}, i, j \in \mathcal{V} \setminus \mathcal{C}_k, i \neq j$
  where $C_k^{ij} = \{h \in \mathcal{C}_k | E_h + SP_{hi} + T_{ij} > L_j$ or $E_h + SP_{hi} > L_i$ or $E_i + T_{ij} + SP_{jh} > L_h$ or $E_j + SP_{jh} > L_h, i, j \in \mathcal{V} \setminus \mathcal{C}_k\}$.
  SP_{ij} represents the shortest traveling time from vertex i to vertex j. When the triangle inequality is not satisfied, the shortest path from i to j can include the visit of other vertices. These constraints detect if a vertex and an arc cannot be simultaneously selected in a feasible solution due to time-windows.

- Generalized subtour elimination inequalities (GSECs).

- Clique inequality. $\sum_{i \in \mathcal{S}} y_i \leq |\mathcal{S}| - 1$.
  If no feasible path passing through all the vertices of a set $\mathcal{S} \subset \mathcal{V}$ exists (due to time-windows), then the number of vertices of $\mathcal{S}$ that can be visited in all the feasible solutions are less than the size of $\mathcal{S}$.

We develop a branch-and-cut algorithm for the GTSPTW. We include at the root node of the branch-and-bound tree all polynomial sets of inequalities while GSEC inequalities and clique inequalities are separated in the course of the algorithm. An initial solution is provided to the algorithm thanks to a heuristic which combines the generation of sequences of clusters and the solution of shortest path problems with resources constraints.

The algorithm is implemented in C++ using Cplex 12.6 and the Concert technology. Preliminary results are obtained on instances generated from GTSP instances. When creating an instance, we guarantee that a feasible solution exists. We set the CPU time limit to 1 hour. Instances with up to 22 clusters and 107 vertices are solved solved to optimality. The average solution time for instances with less than 20 clusters is 19.4 seconds while it is equal to 441.6 seconds for instances with 20 and 22 clusters.

References

[1] M. Joerss, J. Schrder, F. Neuhaus, C. Klink and F. Mann, "Parcel Delivery the Future of Last Mile", Technical Report, McKinsey& Company, 2016.

[2] M. Fischetti, J.J. Salazar González and P. Toth, "A branch-and-cut algorithm for the symmetric generalized traveling salesman problem", *Operations Research* 45(3), 378–394, 1997.

[3] D. Reyes, M. Savelsbergh and A. Toriello, "Vehicle routing with roaming delivery locations", *Transportation Research Part C: Emerging Technologies* 80, 71–91, 2017.

[4] L. Moccia, J.F. Cordeau and G. Laporte, "An incremental tabu search heuristic for the generalized vehicle routing problem with time windows", *Journal of the Operational Research Society* 63, 232–244, 2012.